AMENDMENTS TO THE CLAIMS:

1. (Currently amended) A method of executing a linear algebra subroutine on a machine having at least one floating point unit (FPU) with one or more associated load/store units (LSU) to load data into and out of floating point registers (FRegs) of said FPU, said method comprising:

for an execution code controlling an operation of said floating point unit (FPU) performing a linear algebra subroutine execution, inserting instructions to move data in a contiguous and stride one format into a cache providing data for said FPU for direct loading in a stride one manner into said FPU, so that said LSUs can load said data into said FRegs in an optimal manner before it is scheduled to ~~by~~ be used in said linear algebra subroutine execution, said data being prefetched into said cache from a memory in a register block format predetermined to reduce a number of data streams for a level 3 nested loop matrix-matrix type kernel type operation processing (e.g., level 3 processing) to be three streams and to allow a loading of these streams into said FPU by said LSU,

said register block format comprising a data storage format wherein data is stored in blocks of size p-by-q, where p and q are small integers, meaning that p and q are sufficiently small so that the pieces of these blocks can be fitted into said FRegs, and

wherein said three data streams comprise one stream of data of one matrix of said level 3 processing as considered to be resident in said cache and one stream each for data for two remaining matrix operands of said level 3 processing as residing in a memory or a cache level higher than said cache.

2. (Currently amended) The method of claim 1, wherein said ~~timely~~ moving data is accomplished by scheduling move type instructions into time slots existing in a Level 3 Dense Linear Algebra Subroutine.

3. (Previously presented) The method of claim 1, wherein said linear algebra subroutine comprises a matrix multiplication operation.

4. (Previously presented) The method of claim 1, wherein said linear algebra subroutine comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

5. (Previously presented) The method of claim 1, wherein said linear algebra subroutine invokes a BLAS Level 3 L1 cache kernel.

6. (Currently amended) An apparatus, comprising:

a memory to store matrix data to be used for processing in a linear algebra program;

a floating point unit (FPU) to perform said processing;

a load/store unit (LSU) to load data to be processed by said FPU, said LSU loading said data into a plurality of floating point registers (FRegs); and

a cache to store data from said memory and provide said data to said FRegs,

wherein said matrix data in said memory is moved by having inserted moving instructions for said matrix data to be loaded into said cache prior to a need for said data to be loaded by said LSU into said FRegs for said processing, said data being prefetched into said cache from said memory in a ~~nonstandard~~ format predetermined to reduce a number of data streams for a level 3 linear algebra processing to be three streams and to allow a stride one (e.g., SIMD (single instruction, multiple data) $k > 1$) loading of these streams into said FPU by said LSU,

wherein said ~~nonstandard~~ format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said FRegs, and

wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing is considered to be resident in said cache and two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache.

7. (Original) The apparatus of claim 6, wherein said linear algebra program comprises a matrix multiplication operation.

8. (Previously presented) The apparatus of claim 6, wherein said linear algebra program comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

9. (Previously presented) The apparatus of claim 6, wherein said processing comprises invoking a BLAS Level 3 L1 cache kernel.

10. (Canceled)

11. (Previously presented) The apparatus of claim 6, wherein said moving instructions are inserted into time slots existing in a Level 3 Dense Linear Algebra Subroutine.

12. (Currently amended) A computer-readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of executing linear algebra subroutines on a SIMD (single instruction, multiple data) machine having at least one floating point unit (FPU) with one or more associated load/store units (LSUs) to load data into and out of floating point registers (FRegs) of said FPU by way of a cache, said method comprising:

for an execution code controlling an operation of a floating point unit (FPU) performing a linear algebra subroutine execution, inserting instructions to move data into said cache providing said data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine,

wherein said data is prefetched into said cache from a memory in a ~~nonstandard~~ format predetermined to reduce a number of data streams for a level 3 linear algebra processing to be three streams and to allow a stride one (e.g., SIMD $k > 1$ manner) loading of these streams into said FPU by said LSUs,

wherein said ~~nonstandard~~ format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said FRegs, and

wherein said three data streams comprise one stream as being data of one matrix of said level 3 linear algebra processing ~~is~~ considered to be resident in said cache and two remaining streams, meaning one stream each for remaining two matrix operands of said level 3 linear algebra processing that reside in a memory or a cache level higher than said cache.

13. (Currently amended) The computer-readable storage medium of claim 12, wherein said ~~timely~~ moving data is accomplished by inserting move type instructions into time slots existing in a Level 3 Dense Linear Algebra Subroutine.

14. (Previously presented) The computer-readable storage medium of claim 12, wherein said linear algebra subroutine comprises a matrix multiplication operation.

15. (Previously presented) The computer-readable storage medium of claim 12, wherein said linear algebra subroutine comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

16. (Previously presented) The computer-readable storage medium of claim 12, wherein said linear algebra subroutine invokes a BLAS Level 3 L1 cache kernel.

17. (Currently amended) A method of providing a service involving at least one of solving and applying a scientific/engineering problem, said method comprising at least one of:

using a linear algebra software package that computes one or more matrix subroutines, wherein said linear algebra software package generates an execution code controlling an operation of a floating point unit (FPU) performing a linear algebra subroutine execution, such that instructions are inserted to move data into a cache providing data for said FPU before it is scheduled to be used in the linear algebra subroutine, said data being prefetched from a memory in a ~~nonstandard~~ format predetermined to reduce a number of data streams for a level 3 processing to be three streams and to permit a stride one (e.g., SIMD (single instruction, multiple data) k > 1) loading of these streams into said FPU, wherein said ~~nonstandard~~ format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said FRegs, and

wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing ~~is~~ considered to be resident in said cache and one stream each for two remaining matrix operands of said level 3 linear algebra processing that reside in a memory or a cache level higher than said cache;

providing a consultation for solving a scientific/engineering problem using said linear algebra software package;

transmitting a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result; and

receiving a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result.

18. (Previously presented) The method of claim 17, wherein said linear algebra subroutine comprises an equivalent of a subroutine from LAPACK (Linear Algebra PACKage).

19. (Previously presented) The method of claim 17, wherein said linear algebra subroutine invokes a BLAS Level 3 L1 cache kernel.

20. (Canceled)